

MÉRÉSI UTASÍTÁS

**A DIGITÁLIS JELFELDOLGOZÁS
LABORATÓRIUMI GYAKORLATOKHOZ**

1. MÉRÉS

A mérés tárgya: FIR aluláteresztő szűrő tervezése MATLAB programmal és megvalósítása TEXAS TMS320C5416 DSK-val, lebegőpontos és fixpontos számábrázolással.

A mérés eszközei: Személyi számítógép
TEXAS TMS320C5416 DSK (DSP Starter Kit)
Hálózati adapter

Követelmény: A mérési jegyzőkönyv nyomtatványt a ki kell tölteni és a mérés végén le kell adni.

MÉRÉSI FELADATOK:

1. FIR aluláteresztő szűrő tervezése MATLAB programmal

a, Indítsa el a számítógépet. A Windows7 bejelentkezésekor válassza a **Hallgató** nevű felhasználót. Keresse meg az asztalon és indítsa el a **Virtuális gép**-et. A virtuális gép bejelentkezése után kattintson kettőt a **Virtuális gép - DSP méréshez** sorra. Miután a Windowx XP elindult, tegye teljes képernyőssé. Indítsa el a **Double Commandert**. **Másolja a H:\Dsp\Mérés1 könyvtár tartalmát** (nem a könyvtárat, csak a tartalmát!!) a **C:\Dsplab** könyvtárba (az alkönyvtárakat is). Minimalizálja a Double Commandert.

b, Indítsa el a **MATLAB** programot a **Matlab 5.3** ikonra kattintással. A **help dskfirlof** parancs begépelésével listázza ki a képernyőre a dskfirlof függvény hívási módját és argumentumlistáját.

c, Próbálja ki a **dskfirlof** függvényt néhány különböző fókuszszámú, mintavételi frekvenciájú, határfrekvenciájú és zárósávbeli csillapítású szűrő esetén. Figyelem: a Shannon-törvényt be kell tartani! A függvény kipróbálására legfeljebb 10 percet fordítson. Amikor a futtatás során grafikus képernyő jelenik meg, a továbblépés bármelyik billentyű megnyomásával lehetséges. A futtatáshoz elegendő a help paranccsal listázott egyenlőség jobboldalának begépelése. Egy futtatás befejezése után a \uparrow vagy \downarrow billentyűvel jeleníthetők meg a korábban begépelte parancsok. Az argumentumlista átszerkesztése után az ENTER billentyűvel indíthatunk egy új futtatást. A megismételt futtatások során a grafikus ablak nem ugrik előre automatikusan. Előrehozni az ikonjára kattintással lehet az állapotsoron (a képernyő alján).

Javasolt próbafuttatások:

1. dskfirlof(40,8000,800,65);	4. dskfirlof(60,8000,1000,25);
2. dskfirlof(80,8000,2500,65);	5. dskfirlof(25,8000,1000,25);
3. dskfirlof(180,8000,2500,90);	6. dskfirlof(125,8000,400,50);

(Ne feledkezzen meg a lezáró pontosvesszőről sem!!)

d, Tervezze meg a megvalósítandó FIR aluláteresztő szűrőt az alábbi paraméterekkel:

$$N = 40 \quad F_s = 8000 \text{ Hz} \quad F_c = 2000 \text{ Hz} \quad A_s = 60 \text{ dB}$$

A tervezés a következő parancs begépelésével történik: **dskfirlof(40,8000,2000,60);**

Ábrázolja a jegyzőkönyvben az előírt és a megtervezett szűrő frekvenciamenetét dB-ben (harmadik grafikus képernyő).

Ismételje meg a tervezést, de most a dskfirloq függvénnyel: **dskfirloq(40,8000,2000,60);**

Mindkét függvény fájlba írja a kiszámított együtthatókat, de a dskfirlof lebegőpontos a dskfirloq pedig fixpontos formátumban, C forrásprogramnak megfelelő alakban. Nézze meg a Windows Commander F3 parancsával a C:\Dsplab\MatlabWork könyvtárban a coeffslof.c és a coeffsloq.c fájlok tartalmát. Minimalizálja a Windows Commandert és zárja be a MATLAB programot.

2. A TMS320C5416 DSK-n futtatható DSP program létrehozása és futtatása

a, Csatlakoztassa a DSK modul tápegységét az erősáramú hálózathoz. Várja meg, amíg a felhasználói LED-ek (4 db) villogása megszűnik. Indítsa el a Code Composer Studio-t a **C5416 DSK CCS** ikonra kattintással. Ha a Code Composer Studio részképernyőn jelentkezik be, tegye teljes méretűvé. A **Project** menü **Open** parancsával nyissa meg a **C:\Dsplab\Mérés1.pjt** fájlt. Kattintson a Project ablakban megjelent **Mérés1.pjt** sor előtti + jelre, majd a **Source** sor előtti + jelre. Kattintson kettőt a **Mérés1f.c** feliratra, majd a megjelenő **Mérés1f.c** programlistát tegye teljes méretűvé. Fussa át a szűrőprogram forrásnyelvi listáját és próbálja megérteni, hogyan működik. (A C nyelv ismerői előnyben!) Ha kérdése van, forduljon a mérésvezetőhöz. Ha a forrásprogram "kínaiul" van, folytassa a következő mérési ponttal.

b, Fordítsa le a programot a **Project** menü **Build** parancsával. A fordítás befejezésekor a képernyő alján látható fordítási ablakban az alábbi üzenetnek kell megjelenenni:

Build Complete,

0 Errors, 0 Warnings, 0 Remarks.

Ha a három adat közül valamelyik nem 0, jelezze a mérésvezetőnek. Ha minden rendben van, kattintson a fordítási ablakra az egér **jobb** gombjával és a **Hide** menüponttal rejtse el.

c, A következő feladat a lefordított program **letöltése** a DSK-ba és **futtatása**. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\Mérés1.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával.

d, Minimalizálja a **Code Composer Studio**-t (kattintson az ikonjára a képernyő alján) és indítsa el a **SpectraPro** spektrumanalizátor programot az ikonjára kattintással. Nyomja meg az F8 billentyűt és töltsse be a **Mérés1.cfg** nevű fájlt. Ellenőrizze, hogy a **Plot Top** ablakban látható érték **0.00 dB**, a **Plot Range** ablakban lévő pedig **70.00 dB**. **A SpectraPro program semmilyen beállításán ne módosítson, csak ha erre a mérésvezető külön felszólítja!!** Kattintson a bal felső sarokban a **Run** gombra, várjon addig, amíg az ábrázolt átviteli görbe már nem változik (5-10 sec) majd kattintson a **Stop** gombra. A program hibája miatt a görbe felrajzoltatását az első futtatáskor célszerű megismételni. **Ismételje meg!** A program hangfrekvenciás sweep jelet generál a hangkártya kimenetén és ábrázolja a bemenetre visszaérkező jel spektrumát. Ezzel felvehetjük a kimenet és a bemenet közé kötött négy pólus (jelen esetben a DSK-val megvalósított szűrő) átviteli karakterisztikáját. A karakterisztika felrajzoltatását a **Run** és **Stop** gombok használatával megismételheti, ha szükséges. Ha kész, nyomja meg az **Overlays** felirat alatt található **Set** gombok közül az **1-es** jelűt. **Ne az üres kockába tegyen pipát, hanem a mellette lévő 1-es gombot nyomja meg!**

e, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. A **Project** ablakban kattintson az **egér** jobb gombjával a **Mérés1f.c** sorra. Válassza a **Remove from project** menüpontot. Ezután a **Project** menü **Add Files to Project...** pontjával válassza ki a **C:\Dsplab\Mérés1q.c** fájlt majd kattintson kettőt a **Mérés1q.c** sorra. Ezzel a lebegőpontos ábrázolást használó szűrőprogramot fixpontosra cseréltük. Próbálja meg kitalálni, hogy mi változott a programlistában.

f, Fordítsa le a programot a **Project** menü **Build** parancsával. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\Mérés1.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz és vegye fel a fixpontos szűrő karakterisztikáját is a **Run** és **Stop** gombok használatával. Most ugyanabban a koordináta rendszerben látja mindkét karakterisztikát. Ábrázolja a görbéket a mérési jegyzőkönyvben. Hasonlítsa össze a megtervezett és a lebegőpontosan, ill. fixpontosan megvalósított szűrő frekvenciamenetét és értékelje az eredményt.

i, Az **Overlays** felirat alatt található **On** oszlopban szüntesse meg a kijelölést (vegye ki a „pipát”). **Zárja be** a **SpectraPro** programot. A **Code Composer Studio Debug** menü **Halt** pontjával állítsa le a program futását majd **zárja be** a **Code Composer Studio**-t is. Térjen vissza a **Double Commander**-hez és **törölje** a **Dsplab** könyvtár **tartalmát**. Ürítse ki a lomtárát is. Állítsa le a virtuális gépet. Leállításakor válassza a Power Off lehetőséget. Húzza ki a DSK tápegységét az erősáramú hálózathoz. Állítsa le a **Windows**-t. A **kitöltött** mérési **jegyzőkönyvet** (mérőhelyenként **egy**) adja le a mérésvezetőnek, de **ne felejtse el a nevét ráírni**.

2. MÉRÉS

A mérés tárgya: IIR elliptikus (Cauer) sávszűrő tervezése MATLAB programmal és megvalósítása TEXAS TMS320C5416 DSK-val, lebegőpontos és fixpontos számábrázolással.

A mérés eszközei: Személyi számítógép
TEXAS TMS320C5416 DSK (DSP Starter Kit)
Hálózati adapter

Követelmény: A mérési jegyzőkönyv nyomtatványt a ki kell tölteni és a mérés végén le kell adni.

MÉRÉSI FELADATOK:

1. IIR sávszűrő tervezése MATLAB programmal

a, Indítsa el a számítógépet. A Windows7 bejelentkezésekor válassza a **Hallgató** nevű felhasználót. Keresse meg az asztalon és indítsa el a **Virtuális gép**-et. A virtuális gép bejelentkezése után kattintson kettőt a **Virtuális gép - DSP méréshez** sorra. Miután a Windowx XP elindult, tegye teljes képernyőssé. Indítsa el a **Double Commandert**. **Másolja** a **H:\Dsp\Mérés2** könyvtár **tartalmát** (nem a könyvtárat, csak a **tartalmát!!**) a **C:\Dsplab** könyvtárba (az alkönyvtárakat is). Minimalizálja a Double Commandert.

b, Indítsa el a **MATLAB** programot a **Matlab 5.3** ikonra kattintással. A **help dskirbpf** parancs begépelésével listázza ki a képernyőre a dskirbpf függvény hívási módját és argumentumlistáját.

c, Próbálja ki a **dskirbpf** függvényt néhány különböző fokszerű, mintavételi frekvenciájú, határfrekvenciájú, áteresztősávbeli hullámosságú és zárósávbeli csillapítású szűrő esetén. Figyelem: a Shannon-törvényt be kell tartani! A függvény kipróbálására legfeljebb 10 percet fordítson. Amikor a futtatás során grafikus képernyő jelenik meg, a továbblépés bármelyik billentyű megnyomásával lehetséges. A futtatáshoz elegendő a help paranccsal listázott egyenlőség jobboldalának begépelése. Egy futtatás befejezése után a ↑ vagy ↓ billentyűvel jeleníthetők meg a korábban begépelte parancsok. Az argumentumlista átszerkesztése után az ENTER billentyűvel indíthatunk egy új futtatást. A megismételt futtatások során a grafikus ablak nem ugrik előre automatikusan. Előrehozni az ikonjára kattintással lehet az állapotsoron (a képernyő alján).

Javasolt próbafuttatások: 1. dskirbpf(3,8000,800,2500,1,35); 4. dskirbpf(9,8000,1000,1500,1,70);
2. dskirbpf(5,8000,800,2500,1,35); 5. dskirbpf(4,8000,1300,1500,0.5,50);
3. dskirbpf(6,8000,1000,1500,1,70); 6. dskirbpf(4,8000,1300,1500,3,50);

(Ne feledkezzen meg a lezáró pontosvesszőről sem!!)

d, Tervezze meg a megvalósítandó IIR sávszűrőt az alábbi paraméterekkel:

$N = 6$ $F_s = 8000$ Hz $F_{c1} = 1000$ Hz $F_{c2} = 2000$ Hz $R_p = 0.5$ dB $R_s = 50$ dB

A tervezés a következő parancs begépelésével történik: **dskirbpf(6,8000,1000,2000,0.5,50);**

Ábrázolja a jegyzőkönyvben az előírt és a megtervezett szűrő frekvenciamenetét dB-ben (második grafikus képernyő).

Ismételje meg a tervezést, de most a dskirbpf függvénnyel: **dskirbpf(6,8000,1000,2000,0.5,50);**

A dskirbpf MATLAB függvény a kiszámított coea együtthatókészletet 2^9 -el (512-vel) a coeb együtthatókat 2^{18} -al (262144-el) szorozza és a legközelebbi egész számra kerekíti (lefelé). Az így kvantált együtthatók mindegyike elfér a 16 bites fixpontos ábrázolás -32768 - +32767 tartományában.

Mindkét függvény fájlba írja a kiszámított együtthatókat, de a dskirbpf lebegőpontos a dskirbpf pedig fixpontos formátumban, C forrásprogramnak megfelelő alakban. Nézze meg a Windows Commander F3 parancsával a C:\Dsplab\MatlabWork könyvtárban az acoeffsiirbpf.c és a bcoeffsiirbpf.c valamint az acoeffsiirbpf.c és a bcoeffsiirbpf.c fájl tartalmát. A sávszűrőt mindig aluláteresztőből származtatjuk, ezért a 6-odfokú szűrőhöz a acoeffsiirbpf.c és a bcoeffsiirbpf.c valamint az acoeffsiirbpf.c és a bcoeffsiirbpf.c fájlokban 13-13 együttható tartozik. Minimalizálja a Windows Commandert és zárja be a MATLAB programot.

2. A TMS320C5416 DSK-n futtatható DSP program létrehozása és futtatása

a, Csatlakoztassa a DSK modulhoz a DC adapter csatlakozó dugóját. Várja meg, amíg a felhasználói LED-ek (4 db) villogása megszűnik. Indítsa el a Code Composer Studio-t a **C5416 DSK CCS** ikonra kattintással. Ha a Code Composer Studio részképernyőn jelentkezik be, tegye teljes méretűvé. A **Project** menü **Open** parancsával nyissa meg a C:\Dsplab\Mérés2.pjt fájl. Kattintson a Project ablakban megjelent **Mérés2.pjt** sor előtti + jelre, majd a **Source** sor előtti + jelre. Kattintson kettőt a **Mérés2f.c** feliratra, majd a megjelenő **Mérés2f.c** programlistát tegye teljes méretűvé. Fussa át a szűrőprogram forrásnyelvi listáját és próbálja megérteni, hogyan működik. (A C nyelv ismerői előnyben!) Ha kérdése van, forduljon a mérésvezetőhöz. Ha a forrásprogram "kínaiul" van, folytassa a következő mérési ponttal.

b, Fordítsa le a programot a **Project** menü **Build** parancsával. A fordítás befejezésekor a képernyő alján látható fordítási ablakban az alábbi üzenetnek kell megjelenenni:

```
Build Complete,  
0 Errors, 0 Warnings, 0 Remarks.
```

Ha a három adat közül valamelyik nem 0, jelezze a mérésvezetőnek. Ha minden rendben van, kattintson a fordítási ablakra az egér **jobb** gombjával és a **Hide** menüponttal rejtse el.

c, A következő feladat a lefordított program **letöltése** a DSK-ba és **futtatása**. Töltse le a **File** menü **Load program** parancsával a C:\Dsplab\Debug\Mérés2.out fájl a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával.

d, Minimalizálja a **Code Composer Studio**-t (kattinon az ikonjára a képernyő alján) és indítsa el a **SpectraPro** spektrumanalizátor programot az ikonjára kattintással. Nyomja meg az F8 billentyűt és töltse be a **Mérés2.cfg** nevű fájl. Ellenőrizze, hogy a **Plot Top** ablakban látható érték **10.00 dB**-re, a **Plot Range** ablakban lévő pedig **80.00 dB**. A **SpectraPro** program **semmilyen más beállításán ne módosítson, csak akkor, ha erre a mérésvezető külön felszólítja!!** Kattintson a bal felső sarokban a **Run** gombra, várjon addig, amíg az ábrázolt átviteli görbe már nem változik (5-10 sec) majd kattintson a **Stop** gombra. A program hibája miatt a görbe felrajzoltatását az első futtatáskor célszerű megismételni. Ismétlje meg! A program hangfrekvenciás sweep jelet generál a hangkártya kimenetén és ábrázolja a bemenetre visszaérkező jel spektrumát. Ezzel felvehetjük a kimenet és a bemenet közé kötött négy pólus (jelen esetben a DSK-val megvalósított szűrő) átviteli karakterisztikáját. A karakterisztika felrajzolását a **Run** és **Stop** gombok használatával megismételheti, ha szükséges. Ha kész, nyomja meg az **Overlays** felirat alatt található **Set** gombok közül az **1-es** jelűt. (Ne az üres kockába tegyen pipát, hanem a mellette lévő **1-es** gombot nyomja meg!)

e, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. A **Project** ablakban kattintson az egér jobb gombjával a **Mérés2f.c** sorra. Válassza a **Remove from project** menüpontot. Ezután a **Project** menü **Add Files to Project...** pontjával válassza ki a C:\Dsplab\Mérés2q.c fájl majd kattintson kettőt a **Mérés2q.c** sorra. Ezzel a lebegőpontos ábrázolást használó szűrőprogramot kicseréltük fixpontosra. Fussa át a programlistát és próbálja meg kitalálni, hogy mi változott benne.

f, Fordítsa le a programot a **Project** menü **Build** parancsával. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\Mérés2.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz és vegye fel a fixpontos szűrő karakterisztikáját is a **Run** és **Stop** gombok használatával. Most ugyanabban a koordináta rendszerben látja mindkét karakterisztikát. Ábrázolja a görbékét a mérési jegyzőkönyvben. Hasonlítsa össze a megtervezett és a lebegőpontosan, ill. fixpontosan megvalósított szűrő frekvenciamenetét és értékelje az eredményt.

i, Az **Overlays** felirat alatt található **On** oszlopban szüntesse meg a kijelölést (vegye ki a „pipát”). **Zárja be** a **SpectraPro** programot. A **Code Composer Studio Debug** menü **Halt** pontjával állítsa le a program futását majd **zárja be** a **Code Composer Studio**-t is. Térjen vissza a **Double Commander**-hez és **törölje** a **Dsplab** könyvtár **tartalmát**. Űrítse ki a lomtárat is. Állítsa le a virtuális gépet. Leállításkor válassza a Power Off lehetőséget. Húzza ki a DSK tápegységét az erősáramú hálózathoz. Állítsa le a **Windows**-t. A **kitöltött** mérési **jegyzőkönyvet** (mérőhelyenként **egyet**) adja le a mérésvezetőnek, de **ne felejtse el a nevét ráírni**.

3. MÉRÉS

A mérés tárgya: AM jel, háromszög- fűrész- és négyszögjel generálása TEXAS TMS320C5416 DSK-val és a spektrumuk vizsgálata.

A mérés eszközei: Személyi számítógép
TEXAS TMS320C5416 DSK (DSP Starter Kit)
Hálózati adapter

Követelmény: A mérési jegyzőkönyv nyomtatványt a ki kell tölteni és a mérés végén le kell adni.

MÉRÉSI FELADATOK:

A TMS320C5416 DSK-n futtatható DSP programok létrehozása és futtatása

a, Indítsa el a számítógépet. A Windows7 bejelentkezésekor válassza a **Hallgató** nevű felhasználót. Keresse meg az asztalon és indítsa el a **Virtuális gép**-et. A virtuális gép bejelentkezése után kattintson kettőt a **Virtuális gép - DSP méréshez** sorra. Miután a Windowx XP elindult, tegye teljes képernyőssé. Indítsa el a **Double Commandert**. Másolja a **H:\Dsp\Mérés3** könyvtár **tartalmát** (nem a könyvtárat, csak a **tartalmát!!**) a **C:\Dsplab** könyvtárba (az alkönyvtárakat is). Minimalizálja a Double Commandert.

b, Csatlakoztassa a DSK modulhoz a DC adapter csatlakozó dugóját. Várja meg, amíg a felhasználói LED-ek (4 db) villogása megszűnik. Indítsa el a Code Composer Studio-t a **C5416 DSK CCS** ikonra kattintással. Ha a Code Composer Studio részképernyőn jelentkezik be, tegye teljes méretűvé. A **Project** menü **Open** parancsával nyissa meg a **C:\Dsplab\Mérés3.pjt** fájlt. Kattinon a Project ablakban megjelent **Mérés3.pjt** sor előtti + jelre, majd a **Source** sor előtti + jelre. Kattintson kettőt az **am.c** feliratra, majd a megjelenő **am.c** programlistát tegye teljes méretűvé. Fussa át a program forrásnyelvi listáját és próbálja megérteni, hogyan működik. (A C nyelv ismerői előnyben!) Ha kérdése van, forduljon a mérésvezetőhöz. Ha a forrásprogram "kínaiul" van, folytassa a következő mérési ponttal.

c, Fordítsa le a programot a **Project** menü **Build** parancsával. A fordítás befejezésekor a képernyő alján látható fordítási ablakban az alábbi üzenetnek kell megjelenni:

```
Build Complete,  
0 Errors, 0 Warnings, 0 Remarks.
```

Ha a három adat közül valamelyik nem 0, jelezze a mérésvezetőnek. Ha minden rendben van, kattintson a fordítási ablakra az egér **jobb** gombjával és a **Hide** menüponttal rejtse el.

d, A következő feladat a lefordított program **letöltése** a DSK-ba és **futtatása**. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\Mérés3.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával.

e, Minimalizálja a **Code Compose Studio**-t (kattintson az ikonjára a képernyő alján) és indítsa el a **SpectraPro** programot az ikonjára kattintással. Nyomja meg az F8 billentyűt és töltsse be a **Mérés3.cfg** nevű fájlt. Ellenőrizze, hogy a **Plot Top** ablakban látható érték **0.00 dB**, a **Plot Range** ablakban lévő pedig **70.00 dB**. Ezután kattintson a bal felső sarokban a **Run** gombra, várjon addig, amíg az ábrázolt görbe már nem változik (5-10 sec) majd kattintson a **Stop** gombra. A spektrum felrajzolását a **Run** és **Stop** gombok használatával megismételheti, ha szükséges. A program hibája miatt a görbe felrajzoltatását az első futtatáskor célszerű megismételni. **Ismételje meg!** Tegye teljes méretűvé a **Time Series** ablakot és a nagyító billentyűk használatával vizsgálja meg az időfüggvényt. Tegye ismét részképernyőssé a **Time Series** ablakot és teljes méretűvé a **Spectrum** ablakot. A nagyító gombokkal érje el, hogy az ábra vízszintes tengelye 0 – 2 kHz közé essen. Ábrázolja a spektrumot a mérési jegyzőkönyvben, majd nyomja meg az **OutFull** gombot, és tegye ismét részképernyőssé a **Spectrum** ablakot.

f, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. A **File** menü **Open** parancsával nyissa meg a **saw.c** programot. Fussa át a program forrásnyelvi listáját és próbálja megérteni, hogyan működik. Ezt a későbbi mérési pontokban majd ismétlje meg a **trian.c**, **pulse1_1.c**, **pulse1_3.c** és a **pulse1_4.c** programok esetében is.

g, A **saw.c**, **trian.c**, **pulse1_1.c**, **pulse1_3.c** és a **pulse1_4.c** lefordítása nem szükséges, mert a letölthető, futtatható programok megtalálhatók a **C:\Dsplab\Debug** alkönyvtárban. Töltse le a **File** menü **Load** program parancsával a **C:\Dsplab\Debug\saw.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz. Nyomja meg az **F4** funkcióbillentyűt és a mintavételi frekvenciát állítsa **48000 Hz**-re. Kattintson az **OK** gombra, majd az **OutFull** ikonbillentyűre. A **Plot Top** ablakban látható értéket állítsa **0.00 dB**-re, a **Plot Range** ablakban lévő pedig **75.00 dB**-re. Ezután kattintson a **Run** gombra, várjon addig, amíg az ábrázolt görbe már nem változik (5-10 sec) majd kattintson a **Stop** gombra. Tegye teljes méretűvé a **Time Series** ablakot és a nagyító billentyűk használatával vizsgálja meg az időfüggvényt. Tegye ismét részképernyőssé a **Time Series** ablakot és teljes méretűvé a **Spectrum** ablakot. Ábrázolja a fűrészelj spektrumát 0 – 24 kHz között a mérési jegyzőkönyvben. A nagyító gombokkal érje el, hogy az ábra vízszintes tengelye 0 – 5 kHz közé essen. Ábrázolja a spektrumot a mérési jegyzőkönyvben 0 – 5 kHz között, majd nyomja meg az **OutFull** gombot, és tegye ismét részképernyőssé a **Spectrum** ablakot.

h, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\triang.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz. A **Plot Top** ablakban látható értéket állítsa **10.00 dB**-re, a **Plot Range** ablakban lévő pedig **85.00 dB**-re. Az előzőekben leírt módon vizsgálja meg az időfüggvényt, majd a háromszögjel spektrumát ábrázolja a mérési jegyzőkönyvben 0 – 24 kHz és 0 – 5 kHz tartományban (Teljes képernyős mód!). Végül nyomja meg az **OutFull** gombot, és tegye ismét részképernyőssé a **Spectrum** ablakot.

i, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\pulse1_1.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz. Az előzőekben leírt módon vizsgálja meg az időfüggvényt, majd az 1:1 kitöltésű négyszögjel spektrumát ábrázolja a mérési jegyzőkönyvben 0 – 24 kHz és 0 – 5 kHz tartományban (Teljes képernyős mód!). Végül nyomja meg az **OutFull** gombot, és tegye ismét részképernyőssé a **Spectrum** ablakot.

j, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\pulse1_3.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz. Az előzőekben leírt módon vizsgálja meg az időfüggvényt, majd az 1:3 kitöltésű négyszögjel spektrumát ábrázolja a mérési jegyzőkönyvben 0 – 24 kHz és 0 – 5 kHz tartományban (Teljes képernyős mód!). Végül nyomja meg az **OutFull** gombot, és tegye ismét részképernyőssé a **Spectrum** ablakot.

k, Térjen vissza a **Code Composer Studio**-hoz és a **Debug** menü **Halt** parancsával állítsa le a program futását a DSK-ban. Töltse le a **File** menü **Load program** parancsával a **C:\Dsplab\Debug\pulse1_4.out** fájlt a DSK-ba. Indítsa el a programot a **Debug** menü **Run** parancsával. Térjen vissza a **SpectraPro** programhoz. Az előzőekben leírt módon vizsgálja meg az időfüggvényt, majd az 1:4 kitöltésű négyszögjel spektrumát ábrázolja a mérési jegyzőkönyvben 0 – 24 kHz és 0 – 5 kHz tartományban (Teljes képernyős mód!). Végül nyomja meg az **OutFull** gombot, és tegye ismét részképernyőssé a **Spectrum** ablakot.

l, **Zárja be** a **SpectraPro** programot. A **Code Composer Studio** **Debug** menü **Halt** pontjával állítsa le a program futását majd **zárja be** a **Code Composer Studio**-t is. Térjen vissza a **Double Commander**-hez és **törölje** a **Dsplab** könyvtár **tartalmát**. **Ürítse ki** a **lomtárat** is. Állítsa le a virtuális gépet. Leállításakor válassza a **Power Off** lehetőséget. **Húzza ki** a DSK tápegységét az erősáramú hálózathoz. Állítsa le a **Windows**-t. A **kitöltött** mérési **jegyzőkönyvet** (mérőhelyenként **egyet**) adja le a mérésvezetőnek, de **ne felejtse el a nevét ráírni**.

A DSK-kártya és a Code Composer Studio hibaüzenetei, a hibák elhárítása:

1. File not found – válaszként a **No** gombot kell megnyomni.

2. CPU_1-DSP Device Driver – a **Reset Emulator** gombot kell megnyomni. Utána egy újabb ablakban egy másik üzenet jelenik meg:

Reset Emulator – a **Continue** gombot kell megnyomni. Újabb ablakban újabb üzenet jelenik meg:

Reconnect Emulator – a **Reconnect** gombot kell megnyomni.

Fenti három művelet végrehajtása után a számítógép újracsatlakozik a DSP-kártyához, az utoljára letöltött programot újratölti a DSP memóriájába, de nem indítja el, ezt nekünk kell megtenni, ha szükséges.

```
function [coe]=dskfirlof(N,fs,fc,As);

% [coe]=dskfirlof(N,fs,fc,As);
% A program FIR digitális aluláteresztő szűrő
% együtthatókészletét számítja ki ablak-módszerrel.
%
% N = A szűrő fokszáma
% fs = Mintavételi frekvencia [Hz]
% fc = Határfrekvencia [Hz] (0 < Fc < Fs/2)
% As = Zárósávbeli minimális csillapítás [dB]

fn = 2*(fc/fs);
Us = 1/(10^(As/20));
f = [0 fn fn 1];
H = [1 1 Us Us];
fhz = f*fs/2;

plot(fhz,H), title('Az előírt frekvenciamenet:');
xlabel('Frekvencia (Hz)'), ylabel('Csillapítás')
grid
pause

[coe] = fir1(N-1,fn,chebwin(N,As));

fid = fopen('coeffslof.c','w');
fprintf(fid,'\nconst float filcoeff[%d]',N);
fprintf(fid,'%c\n%c','=','{');
for k = 1:length(coe)-1
    if (mod(k,4)) ~= 0;
        fprintf(fid,'%+1.6e, ',coe(k));
    else
        fprintf(fid,'%+1.6e, \n',coe(k));
    end
end;
fprintf(fid,'%+1.6e \n',coe(length(coe)));
fprintf(fid,'%c%c\n','}',';');
fclose(fid);

areal=fft(coe,1024);
plot(fhz,H,fs*[0:511]/1024,abs(areal(1:512)));
title('Az előírt és a méretezett szűrő frekvenciamenete');
xlabel('Frekvencia (Hz)'), ylabel('Csillapítás')
grid
pause
plot(fs*[0:511]/1024,20*log10(abs(areal(1:512))),fhz,20*log10(H));
title('Frekvenciamenet dB-ben');
xlabel('Frekvencia (Hz)'),ylabel('Csillapítás (dB)')
grid
```

```
function [coe]=dskfirloq(N,fs,fc,As);
% [coe]=dskfirloq(N,fs,fc,As);
% A program FIR digitális aluláteresztő szűrő
% együtthetőkészletét számítja ki ablak-módszerrel.
%
% N = A szűrő fokszáma
% fs = Mintavételi frekvencia [Hz]
% fc = Határfrekvencia [Hz] (0 < Fc < Fs/2)
% As = Zárósávbeli minimális csillapítás [dB]

fn = 2*(fc/fs);
Us = 1/(10^(As/20));
f = [0 fn fn 1];
H = [1 1 Us Us];
fhz = f*fs/2;

plot(fhz,H), title('Az előírt frekvenciamenet:');
xlabel('Frekvencia (Hz)'), ylabel('Csillapítás')
grid
pause

[coeq] = fir1(N-1,fn,chebwin(N,As));
coeq=fix(coeq*(2^15));

fid = fopen('coeffsloq.c','w');
fprintf(fid,'\nconst short filcoeff[%d]',N);
fprintf(fid,'%c\n%c','=','{');
for k = 1:length(coeq)-1
    if (mod(k,5)) ~= 0;
        fprintf(fid,'%d, ',coeq(k));
    else
        fprintf(fid,'%d, \n',coeq(k));
    end
end;
fprintf(fid,'%d\n',coeq(length(coeq)));
fprintf(fid,'%c%c\n','}',';');
fclose(fid);

coer=real(coe);
areal=fft(coe,1024);
plot(fhz,H,fs*[0:511]/1024,abs(areal(1:512)));
title('Az előírt és a méretezett szűrő frekvenciamenete');
xlabel('Frekvencia (Hz)'), ylabel('Csillapítás')
grid
pause
plot(fs*[0:511]/1024,20*log10(abs(areal(1:512))),fhz,20*log10(H));
title('Frekvenciamenet dB-ben');
xlabel('Frekvencia (Hz)'),ylabel('Csillapítás (dB)')
grid
```

```
// *****
// * FIR digitális aluláteresztő szűrő *
// *           Fs = 8 kHz           *
// *   Lebegőpontos szűrőegyütthetők *
// *   Lebegőpontos aritmetika       *
// *****
#include "Mérés1cfg.h"
#include "dsk5416.h"
#include "dsk5416_pcm3002.h"

DSK5416_PCM3002_Config setup =
{
    0x17f, // Reg 0 - Bal csatorna DAC csillapítása
    0x17f, // Reg 1 - Jobb csatorna DAC csillapítása
    0x80,  // Reg 2 - Egyéb vezérlőbitek
    0x0    // Reg 3 - Codec adatformátum
};

void UserTask()
{
    DSK5416_PCM3002_CodecHandle hCodec;

    short out;
    short input[40];
    int filtap=40;
    const float filcoeff[40]=
    {-2.267329e-004, -3.452567e-004, +6.083452e-004, +9.943316e-004, -1.539112e-003,
    -2.284659e-003, +3.279885e-003, +4.582154e-003, -6.259960e-003, -8.397708e-003,
    +1.110423e-002, +1.452816e-002, -1.888666e-002, -2.452155e-002, +3.201779e-002,
    +4.248075e-002, -5.828905e-002, -8.565434e-002, +1.474176e-001, +4.493918e-001,
    +4.493918e-001, +1.474176e-001, -8.565434e-002, -5.828905e-002, +4.248075e-002,
    +3.201779e-002, -2.452155e-002, -1.888666e-002, +1.452816e-002, +1.110423e-002,
    -8.397708e-003, -6.259960e-003, +4.582154e-003, +3.279885e-003, -2.284659e-003,
    -1.539112e-003, +9.943316e-004, +6.083452e-004, -3.452567e-004, -2.267329e-004
    };
    short jobb,bal;
    int i,j;
    float y;

    // Codec start
    hCodec = DSK5416_PCM3002_openCodec(0, &setup);
    DSK5416_PCM3002_setFreq(hCodec, 8000);

    while(1)
    {
        while (!DSK5416_PCM3002_read16(hCodec, &jobb));
        while (!DSK5416_PCM3002_read16(hCodec, &bal));
        input[0]=jobb/2+bal/2;

        y=0;
        for (j=0; j<filtap; j++)
            y += input[j]*filcoeff[j];
        out =(short)y;
        for (i=filtap-1; i!=0; i--)
            input[i]=input[i-1];

        while (!DSK5416_PCM3002_write16(hCodec, out));
        while (!DSK5416_PCM3002_write16(hCodec, out));

    }
}

void main()
{
    DSK5416_init();
}
```

```
// *****
// *   FIR digitális aluláteresztő szűrő   *
// *           Fs = 8 kHz                   *
// *   Fixpontos (kvantált) szűrőegyütthetők *
// *           Fixpontos aritmetika         *
// *****
#include "Mérés1cfg.h"
#include "dsk5416.h"
#include "dsk5416_pcm3002.h"

DSK5416_PCM3002_Config setup =
{
    0x17f, // Reg 0 - Bal csatorna DAC csillapítása
    0x17f, // Reg 1 - Jobb csatorna DAC csillapítása
    0x80,  // Reg 2 - Egyéb vezérlőbitek
    0x0    // Reg 3 - Codec adatformátum
};

void UserTask()
{
    DSK5416_PCM3002_CodecHandle hCodec;

    short out;
    int filtap=40;
    short input[40];
    const short filcoeff[40]=
    {-7, -11, 19, 32, -50,
     -74, 107, 150, -205, -275,
     363, 476, -618, -803, 1049,
     1392, -1910, -2806, 4830, 14725,
     14725, 4830, -2806, -1910, 1392,
     1049, -803, -618, 476, 363,
     -275, -205, 150, 107, -74,
     -50, 32, 19, -11, -7
    };
    short jobb,bal;
    int i,j;
    long y;

    // Codec start
    hCodec = DSK5416_PCM3002_openCodec(0, &setup);
    DSK5416_PCM3002_setFreq(hCodec, 8000);

    while(1)
    {
        while (!DSK5416_PCM3002_read16(hCodec, &jobb));
        while (!DSK5416_PCM3002_read16(hCodec, &bal));
        input[0]=jobb/2+bal/2;

        y=0;
        for (j=0; j<filtap; j++)
            y += input[j]*filcoeff[j];
        out =(short) (y>>15);
        for (i=filtap-1; i!=0; i--)
            input[i]=input[i-1];

        while (!DSK5416_PCM3002_write16(hCodec, out));
        while (!DSK5416_PCM3002_write16(hCodec, out));
    }
}

void main()
{
    DSK5416_init();
}
```

```
function [coeb,coea]=dskiiirbpf(N,fs,fc1,fc2,rp,rs);

% [coeb,coea]=dskiiirbpf(N,fs,fc,As);
% A program elliptikus (Cauer) digitális sávszűrő
% együtthatókészletét számítja ki.
%
% N = A szűrő fokszáma
% fs = Mintavételi frekvencia [Hz]
% fc1,fc2 = Határfrekvenciák [Hz] (0 < fc1 < fc2 < fs/2)
% rp = Áteresztősávbeli maximális hullámosság [dB]
% rs = Zárósávbeli minimális csillapítás [dB]

fn1 = 2*(fc1/fs);
fn2 = 2*(fc2/fs);
fn = [fn1 fn2];
Us = 1/(10^(rs/20));
f = [0 fn1 fn1 fn2 fn2 1];
H = [Us Us 1 1 Us Us];
fhz = f*fs/2;

plot(fhz,H), title('Az előírt frekvenciamenet:');
xlabel('Frekvencia (Hz)'), ylabel('Csillapítás')
grid
pause

[coeb,coea] = ellip(N,rp,rs,fn);

fid = fopen('bcoeffsiirbpf.c','w');
fprintf(fid,'\nconst float filcoeff[%d]',2*N+1);
fprintf(fid,'%c\n%c','=','{');
for k = 1:length(coeb)-1
    if (mod(k,4)) ~= 0;
        fprintf(fid,'%+1.6e, ',coeb(k));
    else
        fprintf(fid,'%+1.6e, \n',coeb(k));
    end
end;
fprintf(fid,'%+1.6e \n',coeb(length(coeb)));
fprintf(fid,'%c%c\n','}',';');
fclose(fid);

fid = fopen('acoeffsiirbpf.c','w');
fprintf(fid,'\nconst float filcoeff[%d]',2*N+1);
fprintf(fid,'%c\n%c','=','{');
for k = 1:length(coeb)-1
    if (mod(k,4)) ~= 0;
        fprintf(fid,'%+1.6e, ',coea(k));
    else
        fprintf(fid,'%+1.6e, \n',coea(k));
    end
end;
fprintf(fid,'%+1.6e \n',coea(length(coea)));
fprintf(fid,'%c%c\n','}',';');
fclose(fid);

n=8192;
ff=fs/(2*n)*(0:n-1);
areal=abs(freqz(coeb,coea,n));
plot(ff(2:n),20*log10(areal(2:n)),fhz,20*log10(H));
title('Frekvenciamenet dB-ben');
xlabel('Frekvencia (Hz)'),ylabel('Csillapítás (dB)')
grid
```

```
function [coeb,coea]=dskiiirbpf(N,fs,fc1,fc2,rp,rs);

% [coeb,coea]=dskiiirbpf(N,fs,fc,As);
% A program elliptikus (Cauer) digitális sávszűrő
% együtthatókészletét számítja ki.
%
% N = A szűrő fokszáma
% fs = Mintavételi frekvencia [Hz]
% fc1,fc2 = Határfrekvenciák [Hz] (0 < fc1 < fc2 < fs/2)
% rp = Áteresztősávbeli maximális hullámosság [dB]
% rs = Zárósávbeli minimális csillapítás [dB]

fn1 = 2*(fc1/fs);
fn2 = 2*(fc2/fs);
fn = [fn1 fn2];
Us = 1/(10^(rs/20));
f = [0 fn1 fn1 fn2 fn2 1];
H = [Us Us 1 1 Us Us];
fhz = f*fs/2;

plot(fhz,H), title('Az előírt frekvenciamenet:');
xlabel('Frekvencia (Hz)'), ylabel('Csillapítás')
grid
pause

[coeb,coea] = ellip(N,rp,rs,fn);
coeaq=fix(coea*(2^9));
coebq=fix(coeb*(2^18));

fid = fopen('bcoeffsiirbpq.c','w');
fprintf(fid,'\nconst short filcoeff[%d]',2*N+1);
fprintf(fid,'%c\n%c','=','{');
for k = 1:length(coebq)-1
    if (mod(k,5)) ~= 0;
        fprintf(fid,'%d, ',coebq(k));
    else
        fprintf(fid,'%d, \n',coebq(k));
    end
end;
fprintf(fid,'%d\n',coebq(length(coebq)));
fprintf(fid,'%c%c\n','}',';');
fclose(fid);

fid = fopen('acoeffsiirbpq.c','w');
fprintf(fid,'\nconst short filcoeff[%d]',2*N+1);
fprintf(fid,'%c\n%c','=','{');
for k = 1:length(coeaq)-1
    if (mod(k,5)) ~= 0;
        fprintf(fid,'%d, ',coeaq(k));
    else
        fprintf(fid,'%d, \n',coeaq(k));
    end
end;
fprintf(fid,'%d\n',coeaq(length(coeaq)));
fprintf(fid,'%c%c\n','}',';');
fclose(fid);

n=8192;
ff=fs/(2*n)*(0:n-1);
areal=abs(freqz(coeb,coea,n));
plot(ff(2:n),20*log10(areal(2:n)),fhz,20*log10(H));
title('Frekvenciamenet dB-ben');
xlabel('Frekvencia (Hz)'),ylabel('Csillapítás (dB)')
grid
```

```
// *****  
// * Elliptikus (Cauer) IIR digitális szűrő *  
// *           Fs = 8 kHz           *  
// *   Lebegőpontos szűrőgyűtthetők *  
// *   Lebegőpontos aritmetika       *  
// *****  
  
#include "Mérés2cfg.h"  
#include "dsk5416.h"  
#include "dsk5416_pcm3002.h"  
  
DSK5416_PCM3002_Config setup =  
{  
    0x1fff, // Reg 0 - Bal csatorna DAC csillapítása  
    0x1fff, // Reg 1 - Jobb csatorna DAC csillapítása  
    0x80,   // Reg 2 - Egyéb vezérlőbitek  
    0x0    // Reg 3 - Codec adatformátum  
};  
  
void UserTask()  
{  
    DSK5416_PCM3002_CodecHandle hCodec;  
  
    const float coea[13]=  
    {+1.000000e+000, -4.211564e+000, +1.169972e+001, -2.262576e+001,  
    +3.498358e+001, -4.311250e+001, +4.432915e+001, -3.725833e+001,  
    +2.611079e+001, -1.454557e+001, +6.466854e+000, -1.987426e+000,  
    +4.073528e-001  
    };  
  
    const float coeb[13]=  
    {+9.766346e-003, -2.869099e-002, +4.872126e-002, -6.894890e-002,  
    +9.053923e-002, -9.801966e-002, +9.723830e-002, -9.801966e-002,  
    +9.053923e-002, -6.894890e-002, +4.872126e-002, -2.869099e-002,  
    +9.766346e-003  
    };  
  
    short out;  
    float outy[13]={0,0,0,0,0,0,0,0,0,0,0,0,0};  
    short input[13]={0,0,0,0,0,0,0,0,0,0,0,0,0};  
    short jobb,bal;  
    int j;  
    float x,y,ysmp;  
  
    // Codec start  
    hCodec = DSK5416_PCM3002_openCodec(0, &setup);  
    DSK5416_PCM3002_setFreq(hCodec, 8000);  
  
    while(1)  
    {  
        while (!DSK5416_PCM3002_read16(hCodec, &jobb));  
        while (!DSK5416_PCM3002_read16(hCodec, &bal));  
        input[0]=jobb/2+bal/2;  
  
        for (j=12; j!=0; j--) //korábbi kimeneti minták  
            outy[j]=outy[j-1]; //bufferének shiftelése  
        x=0;  
        for (j=0; j<13; j++) //első szumma  
            x+=coeb[j]*input[j];  
        y=0;  
        for (j=1; j<13; j++) //második szumma  
            y+=coea[j]*outy[j];  
        ysmp=x-y; //a különbségük  
        outy[0]=ysmp;  
        out=(short)ysmp;
```



```

        for (j=12; j!=0; j--) //korábbi bimeneti minták
            input[j]=input[j-1]; //bufferének shiftelése

        while (!DSK5416_PCM3002_writel6(hCodec, out));
        while (!DSK5416_PCM3002_writel6(hCodec, out));
    }
}

void main()
{
DSK5416_init();
}

```

A TMS320C5416 DSK-n futó Mérés2q.c program listája:

(2.mérés)

```

// *****
// * Elliptikus (Cauer) IIR digitális szűrő *
// *           Fs = 8 kHz           *
// * Fixpontos (kvantált) szűrőegyütthetők *
// *           Fixpontos aritmetika   *
// *****

#include "Mérés2cfg.h"
#include "dsk5416.h"
#include "dsk5416_pcm3002.h"

DSK5416_PCM3002_Config setup =
{
    0x1ff, // Reg 0 - Bal csatorna DAC csillapítása
    0x1ff, // Reg 1 - Jobb csatorna DAC csillapítása
    0x80,  // Reg 2 - Egyéb vezérlőbitek
    0x0    // Reg 3 - Codec adatformátum
};

void UserTask()
{
DSK5416_PCM3002_CodecHandle hCodec;

const short coea[13]=
{512, -2156, 5990, -11584, 17911,
-22073, 22696, -19076, 13368, -7447,
3311, -1017, 208
};
const float coeb[13]=
{2560, -7521, 12771, -18074, 23734,
-25695, 25490, -25695, 23734, -18074,
12771, -7521, 2560
};

short out;
short outy[13]={0,0,0,0,0,0,0,0,0,0,0,0,0};
short input[13]={0,0,0,0,0,0,0,0,0,0,0,0,0};
short jobb,bal;
int j;
long x,y,ysmp;

// Codec start
hCodec = DSK5416_PCM3002_openCodec(0, &setup);
DSK5416_PCM3002_setFreq(hCodec, 8000);

while(1)
{
    while (!DSK5416_PCM3002_read16(hCodec, &jobb));
    while (!DSK5416_PCM3002_read16(hCodec, &bal));
    input[0]=jobb/2+bal/2;
}

```

```

    for (j=12; j!=0; j--)          //korábbi kimeneti minták
        outy[j]=outy[j-1];      //bufferének shiftelése
    x=0;
    for (j=0; j<13; j++)          //első szumma
        x+=coeb[j]*input[j];
    y=0;
    for (j=1; j<13; j++)          //második szumma
        y+=coea[j]*outy[j];
    ysmpt=(x/262144)-(y/512);     //a különbségük
    outy[0]=(short)ysmpt;
    out=(short)ysmpt;

    for (j=12; j!=0; j--)          //korábbi bemeneti minták
        input[j]=input[j-1];    //bufferének shiftelése

    while (!DSK5416_PCM3002_writel6(hCodec, out));
    while (!DSK5416_PCM3002_writel6(hCodec, out));
}
}

void main()
{
DSK5416_init();
}

```

```

/*****
 *          AM generátor          *
 *          Fs = 48 kHz          *
 *          Fvívő = 1 kHz Fmod = 250 Hz *
 *****/
#include "Mérés3cfg.h"
#include "dsk5416.h"
#include "dsk5416_pcm3002.h"
#include <math.h>
#define CARRIER_TABLE_SIZE 48
#define MOD_TABLE_SIZE 192
#define PI ((double)3.1415927)
#define CARRIER_MAX 0x3FFE
#define MOD_MAX 0x17FE

int carriertable[CARRIER_TABLE_SIZE];
int modtable[MOD_TABLE_SIZE];

DSK5416_PCM3002_Config setup =
{
    0x13f, // Reg 0 - Bal csatorna DAC csillapítása
    0x13f, // Reg 1 - Jobb csatorna DAC csillapítása
    0x0,   // Reg 2 - Egyéb vezérlőbitek
    0x0    // Reg 3 - Codec adatformátum
};

void UserTask()
{
    DSK5416_PCM3002_CodecHandle hCodec;
    int j, sample, sample1, sample2;
    int i;
    double increment= 0;
    double radian = 0;
    // A codec indítása
    hCodec = DSK5416_PCM3002_openCodec(0, &setup);
    // A szinusztáblázatok feltöltése
    increment = (PI * 2) / CARRIER_TABLE_SIZE;
    for (i = 0; i < CARRIER_TABLE_SIZE; i++)
    {
        carriertable[i] = (int)(sin(radian) * CARRIER_MAX);
        radian += increment;
    }
    increment = (PI * 2) / MOD_TABLE_SIZE;
    radian = 0;
    for (i = 0; i < MOD_TABLE_SIZE; i++)
    {
        modtable[i] = (int)(sin(radian) * MOD_MAX);
        radian += increment;
    }
    // AM generáló program végtelen ciklusban
    while(1)
    {
        for (j = 0; j < MOD_TABLE_SIZE; j++)
        {
            sample1 = modtable[j];
            sample2 = carriertable[j % CARRIER_TABLE_SIZE];
            sample = (int)((float)sample1 * ((float)sample2)/((float)CARRIER_MAX));
            while (!DSK5416_PCM3002_write16(hCodec, sample/8+sample2/20));
            while (!DSK5416_PCM3002_write16(hCodec, sample/8+sample2/20));
        }
    }
}

void main()
{
    DSK5416_init();
}

```

A TMS320C5416 DSK-n futó saw.c, trian.c, pulse1 2.c, pulse1 3.c és pulse1 4.c programok listája: (3.mérés)

A programok azonosak, különbség csak a fejlécben és a bekeretezett részben van. A bekeretezett rész változatait külön megadtuk.

```
// *****
// *      Fűrészjel generátor      *
// *          Fs = 48 kHz          *
// *          F = 250 Hz           *
// *****

#include "Mérés3cfg.h"
#include "dsk5416.h"
#include "dsk5416_pcm3002.h"
#include <math.h>

#define SIGNAL_TABLE_SIZE 192
#define SIGNAL_MAX        0x04FE

int signaltable[SIGNAL_TABLE_SIZE];

DSK5416_PCM3002_Config setup =
{
    0x13f, // Reg 0 - Bal csatorna DAC csillapítása
    0x13f, // Reg 1 - Jobb csatorna DAC csillapítása
    0x0,   // Reg 2 - Egyéb vezérlőbitek
    0x0    // Reg 3 - Codec adatformátum
};

void UserTask()
{
    DSK5416_PCM3002_CodecHandle hCodec;
    int i,j,sample;

    // A codec indítása
    hCodec = DSK5416_PCM3002_openCodec(0, &setup);

    // A fűrészjel táblázat feltöltése
    for (i = 0; i < SIGNAL_TABLE_SIZE; i++)
        signaltable[i] = SIGNAL_MAX-(i * (SIGNAL_MAX/SIGNAL_TABLE_SIZE))*2;

    // Fűrészjel generálása végtelen ciklusban
    while(1)
    {
        for (j = 0; j < SIGNAL_TABLE_SIZE; j++)
        {
            sample = signaltable[j];
            while (!DSK5416_PCM3002_write16(hCodec, sample));
            while (!DSK5416_PCM3002_write16(hCodec, sample));
        }
    }

}

void main()
{
    DSK5416_init();
}
```

Az előző oldalon bekeretezett résznek megfelelő programrészletek a különböző jeleket generáló programok esetén:

Háromszögjel:

```
// A háromszögjel táblázat feltöltése
for (i = 0; i < (SIGNAL_TABLE_SIZE/2); i++)
{
    signaltable[i] = i * (2*SIGNAL_MAX/SIGNAL_TABLE_SIZE);
    signaltable[SIGNAL_TABLE_SIZE-i-1] = i * (2*SIGNAL_MAX/SIGNAL_TABLE_SIZE);
}

// Háromszögjel generálása végtelen ciklusban
```

1:1 kitöltésű négyszögjel:

```
// A négyszögjel táblázat feltöltése
for (i = 0; i < SIGNAL_TABLE_SIZE/2; i++)
{
    signaltable[i] = SIGNAL_MAX;
    signaltable[i+SIGNAL_TABLE_SIZE/2] = -SIGNAL_MAX;
}

// Négyszögjel generálása végtelen ciklusban
```

1:3 kitöltésű négyszögjel:

```
// A négyszögjel táblázat feltöltése
for (i = 0; i < SIGNAL_TABLE_SIZE/3; i++)
    signaltable[i] = SIGNAL_MAX;
for (i = SIGNAL_TABLE_SIZE/3; i < SIGNAL_TABLE_SIZE; i++)
    signaltable[i] = -SIGNAL_MAX;

// Négyszögjel generálása végtelen ciklusban
```

1:4 kitöltésű négyszögjel:

```
// A négyszögjel táblázat feltöltése
for (i = 0; i < SIGNAL_TABLE_SIZE/4; i++)
    signaltable[i] = SIGNAL_MAX;
for (i = SIGNAL_TABLE_SIZE/4; i < SIGNAL_TABLE_SIZE; i++)
    signaltable[i] = -SIGNAL_MAX;

// Négyszögjel generálása végtelen ciklusban
```